

# Wound measurement by RGB-D camera

Damir Filko<sup>1</sup> · Robert Cupec<sup>1</sup> · Emmanuel Karlo Nyarko<sup>1</sup>

Received: 28 March 2017 / Revised: 1 December 2017 / Accepted: 17 January 2018 / Published online: 30 March 2018  
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

## Abstract

The robot and computer vision community has seen a lot of novelties developed in the past few years as a result of the appearance of cheap RGB-D sensors spearheaded by the Kinect sensor. In this paper, the feasibility of using an RGB-D camera in detecting, segmenting, reconstructing and measuring chronic wounds in 3D is explored. The wound is detected by implementing nearest-neighbor approach on color histograms generated from the image. The proposed wound segmentation procedure extracts the wound contour using visual and geometrical information of the surface. A procedure comparable to KinectFusion is used for the 3D reconstruction of the wound. In order to achieve real-time performance, the whole system is realized in CUDA. The resulting system provides an accurate colored 3D model of the segmented wound and enables the user to determine the volume, area and perimeter of the wound, thereby aiding in the selection of a suitable therapy. The developed system is experimentally evaluated using the Seymour II wound care model by VATA Inc.

**Keywords** Chronic wound · Detection · 3D reconstruction · Segmentation · Measurement · RGB-D camera

## 1 Introduction

Chronic wounds heal very slowly and the healing process may further be prolonged if an ineffective treatment is used. Clinicians need an objective wound characterization method to decide whether the current treatment is adequate or requires modifications. Accurate wound measurement is an important task in chronic wound treatment, because changes in the physical parameters of the wound are indicators of the healing progress.

Wound measurement methods currently in use are mostly based on simple approaches for area measurement, using rulers and transparency tracing, or photography-based non-contact systems. Such methods give a rough estimate of the

total wound area and the evaluation depends on human experience; it is not objective and not eligible for validation of the wound healing process.

The first step in creating a precise non-contact measurement system is having the ability to reconstruct the wound as a 3D model. This would facilitate complex analysis or measurement of wounds which is otherwise awkward and painful for the patient or inconvenient for clinicians. Furthermore, the reconstructed 3D model could be used for collaboration between clinicians and telemedicine.

Non-contact wound measurement and 3D reconstruction systems have been developed before, utilizing lasers or photography with multiple view geometry algorithms. This results in very expensive systems in the case of the former, or less precise and noise prone systems in the case of the latter. With the advent of inexpensive RGB-D cameras, the computer vision community has gained an easier way to innovate and create applications in many fields including medicine.

In this paper, we explore possibilities of using RGB-D cameras in a system for detection, 3D reconstruction and segmentation of chronic wounds. Three different RGB-D cameras are used: Orbbec Astra S, PrimeSense Carmine 1.09 and Microsoft Kinect v2. The developed system automatically detects wounds by analyzing image blocks according to color histogram similarity using a nearest-neighbor approach. The only manual interaction required from the user is to select

---

This work was supported by the Josip Juraj Strossmayer University of Osijek, under Grant No. IZIP-2014-70.

✉ Damir Filko  
damir.filko@etfos.hr

Robert Cupec  
robert.cupec@etfos.hr

Emmanuel Karlo Nyarko  
nyarko@etfos.hr

<sup>1</sup> Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, Kneza Trpimira 2B, 31000 Osijek, Croatia

a particular wound to be analyzed in the case where multiple wounds are detected. The 3D reconstruction is similar to the KinectFusion approach [1], where Iterative Closest Point (ICP) algorithm [2] is used for determining rigid body transformation. Color-enhanced Truncated Signed Distance Function (TSDF) is applied for scene fusion and the Marching cubes algorithm is used for creating a surface mesh. The wound segmentation algorithm is driven by changes in surface geometry and color with the main goal of separating the wound tissue from the surrounding healthy tissue (skin). Furthermore, except for segmentation, the entire system is implemented in CUDA, which enables real-time operation. The proposed system provides a high-resolution 3D colored model of the wound and its surrounding tissue suitable for further analysis. Analysis implemented in the system includes measurement of wound perimeter, area and volume, which clinicians could use to develop more responsive therapy and treatment for chronic wounds. The 3D models could also be efficiently used for telemedicine, thereby improving clinician collaboration and enhancing the treatment of patients.

The major contribution of this paper is a thorough analysis of the applicability of popular RGB-D cameras in a KinectFusion like system in a medical application for the purpose of measuring chronic wound parameters such as perimeter, area and volume. The technical contribution of this paper is a novel wound analysis system based on the following solutions, which to the best of our knowledge have not been used in existing systems developed for the considered application: (i) 3D reconstruction by fusion of multiple RGB-D images; (ii) wound detection based on nearest-neighbor classifier; (iii) wound segmentation using local surface curvature and color.

The rest of the paper is structured as follows. In Sect. 2, a short survey of related research is given. Section 3 provides an overview of considered RGB-D cameras, while Sect. 4 presents the detection subsystem and Sect. 5 elaborates the 3D reconstruction subsystem. Section 6 gives details on the segmentation subsystem and Sect. 7 explains the methods used for measurement of wound perimeter, area and volume. Section 8 presents and explains experiments conducted in order to determine the measurement accuracy and repeatability. Finally, this paper is concluded with Sect. 9, which comments on the results and future work.

## 2 Related research

A lot of published research focuses on wound assessment by measuring physical attributes such as depth, perimeter, area and volume. Wound measurement is typically divided into two main approaches: contact and non-contact. Contact-based methods require the measurements to be taken directly

with wound contact [3,4]. Rulers or transparent films are used in such situations for measuring depth, perimeter and area, while fluid or similar material is used for volume measurement. The main disadvantages of using contact-based methods are imprecision and subjective interpretation. Contact-based methods also increase the possibility of wound infections and are typically not comfortable for patients.

Non-contact methods require a recording device, whereby a standard camera is typically used. For example, in [5] the wound is manually marked and tissue type percentages calculated, while area and perimeter measurements require that the ruler be visible in the image. Newer systems which use standard camera automatically segment the wound [6,7]. It is generally accepted that image-based methods have a maximum error of area measurement of about 10% depending on the wound location and the camera angle. This fact is experimentally proven in several studies including [8] where the authors also conclude that image-based measurement is a viable alternative to contact-based methods such as VisiTrak.

Other wound assessment systems use 3D reconstruction to enhance image-based methods and gain more precise measurement. Therefore, using multiple view geometry algorithms with standard cameras is a typical approach. In [9], the authors use two wound images taken from different angles to generate a 3D mesh model. Because of the technology and algorithms used, the resulting 3D mesh has a low resolution. Ekyona, a system similar in principle and commercially available, used in the study by [10] generates more dense 3D representation. Besides being very expensive, Ekyona has limitations regarding the size and position of the wound it can reconstruct because the whole wound needs to be visible in a single frame. Ekyona also requires the use of a disposable optical marker which must be located in the image. Besides Ekyona, there are other commercially available 3D wound measurement systems based on stereo vision such as MAVIS and 3D LifeViz, which are equally expensive.

Lasers are also commonly used for 3D reconstruction in medical research, where a laser line projection sensor calibrated with an RGB camera can produce precise and colored 3D reconstructions. One of the first studies which used such a system was Derma [11], where the authors used the expensive Minolta VI910 scanner. Other similar research also used a combination of laser and RGB camera [12,13]. Such systems have proven to be very precise, but are also not easy to use and very expensive. Furthermore, they also share the limitation that stereo vision systems have: the whole wound must be visible in one frame. One commercially available and relatively costly system for wound measurement which uses laser technology is SilhouetteStar by Aranz Medical. Unfortunately, the system only provides measurement results and does not provide a reconstructed 3D model which the user can further analyze.

**Table 1** RGB-D camera specifications

Name	Cost	Min/max range	Field of view ( $H \times V$ )	Depth image resolution	Depth resolution	Color image resolution	Power supply	Weight (kg)
PrimeSense Carmine 1.09	n/a	0.35 m/1.4 m	57.5 × 45	640 × 480	1 mm/100 μm	640 × 480	USB 2.0	0.23
Orbbec Astra S	150\$	0.35 m/2.5 m	60 × 49.5	640 × 480	1 mm	640 × 480	USB 2.0	0.3
Microsoft Kinect v2	100\$ (adapter 40\$)	0.5 m/2.5 m	70 × 60	512 × 424	1 mm	1920 × 1080	Separate adapter, USB 3.0	1.4

RGB-D camera is also used in one commercial system for wound measurement but is not yet available to the general public. InSight by eKare uses a technology similar to the original Microsoft Kinect and is used in the study by [14] for comparison in measurement performance with SilhouetteStar. The measurement shows comparable results except in wound analysis time which takes 45s for InSight and approximately 2 min for SilhouetteStar to produce results. The new InSight system also has the limitation regarding the visibility of the whole wound in a single frame [15].

As it can be seen, most of the systems have limitations regarding precision, ease of use and price, the latter of which limits the widespread use of most systems. However, the biggest limitation in all presented systems is that the whole wound must be visible in a single view. This constraint limits the wounds size and location. The 3D wound reconstruction system presented in this paper does not have such a constraint. Since it is based on scene fusion, the wound can be located on a larger area and on curved surfaces. Furthermore, inexpensive RGB-D cameras are used which cost about \$150 which should make the system available to a much wider user base. Preliminary results of the developed systems performance were published at the conferences [16,17]. This article presents a complete description of developed system including all relevant equations, pseudocodes and examples. Furthermore, research presented in this paper uses two other RGB-D cameras besides the Kinect v2 which was used in the preliminary studies. Also, an in-depth analysis of the detection subsystem was conducted and a more comprehensive database was developed.

### 3 RGB-D cameras and their characteristics

Commercially available and inexpensive RGB-D cameras have been a resounding success in computer and robot vision applications since the appearance of the original Microsoft Kinect in 2010. Besides Microsoft Kinect, other low cost RGB-D cameras have appeared on the market, such as PrimeSense Carmine, Asus Xtion and Orbbec Astra. In the meantime, PrimeSense has been acquired by Apple and products based on their technology such as Carmine and Xtion

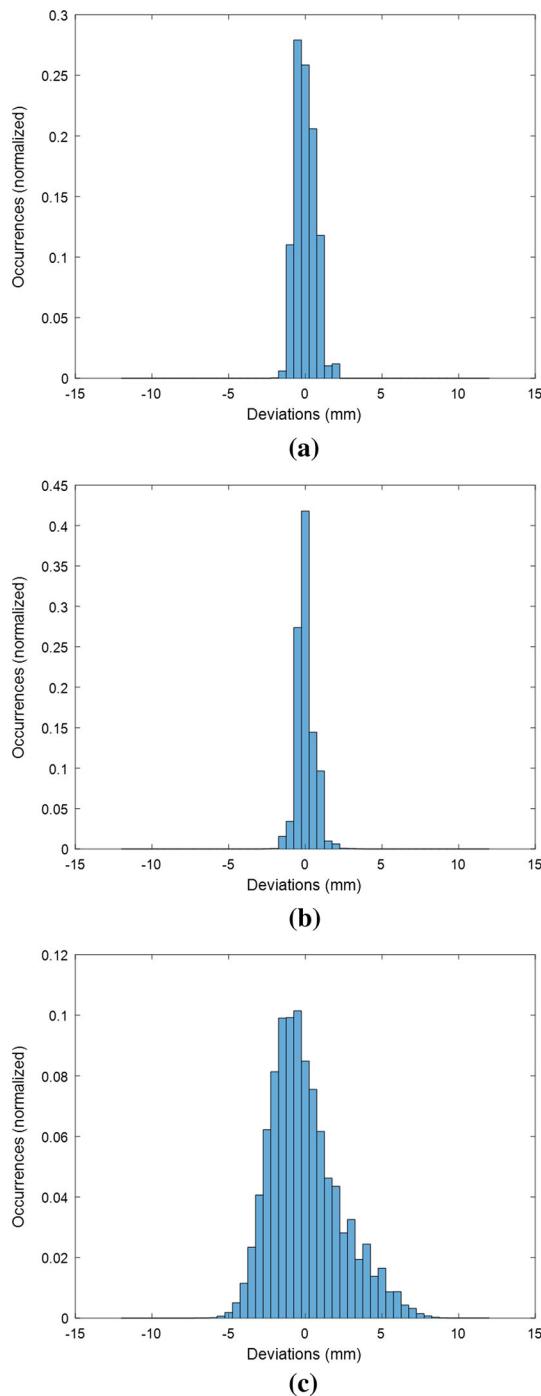
have been hard to find. In 2014, Microsoft introduced Kinect v2 based on the time-of-flight principle as opposed to structured light in the original Kinect. A thorough analysis of Kinect v2 camera capabilities is available in [18]. In 2016, Orbbec presented a new RGB-D camera based on structured light principle similar to the original Kinect. In recent years, Intel has also presented an RGB-D technology called RealSense based on structured light principle. Cameras based on RealSense technology are available as development kit, but we were not able to acquire one during the project. They are also integrated into laptop and tablet devices.

In this article, the applicability of three different RGB-D cameras in a medical application for detection, 3D reconstruction and measuring of chronic wounds is investigated. Table 1 lists basic specifications of the three considered cameras.

As it can be seen, only PrimeSense sensor is currently unavailable and reasons for that is mentioned earlier. All three cameras are relatively easy to handle, although Kinect v2 is the bulkiest and heaviest. Kinect v2 is also the only one that requires USB 3.0 port to function properly. Astra and Carmine cameras support USB 2.0 and 3.0 standard. Kinect v2 and Astra have proprietary API, but Astra also supports open source OpenNI 2 API which Carmine uses as well. Therefore, Astra and Carmine use the same code path within the developed system.

The considered cameras have different methods of operation. Although Astra and Carmine share the same basic principle, they have different responses to environmental conditions such as sensitivity to illumination, surface reflectivity and viewing angle. In order to obtain an insight into the measurement noise of the three considered cameras, we performed an experiment in which an image of a planar wall surface is acquired by all three cameras, a central region of 200 × 200 pixels is extracted and least squares fitting of a plane to the extracted point set is performed. Figure 1 shows the distribution of deviations from the plane. As it can be seen, Kinect v2 time-of-flight camera has considerably more noise in depth measurement than Astra and Carmine cameras which use the structured light principle.

All three cameras support hardware (within the camera) or software (within the API) registration between color and



**Fig. 1** Residual distances between points and a fitted plane for **a** Orbbee Astra S ( $\mu = 4.32 \times 10^{-15}$ ,  $\sigma = 0.64$ ), **b** PrimeSense Carmine 1.09 ( $\mu = 7.12 \times 10^{-15}$ ,  $\sigma = 0.54$ ), **c** Microsoft Kinect v2 ( $\mu = 1.85 \times 10^{-15}$ ,  $\sigma = 2.29$ )

depth images. The images in Fig. 2 show color to depth registration as well as point cloud screenshots.

It can be noticed that the factory registration for all considered cameras has various degrees of success. Kinect v2 has clearly better registration than Astra S and Carmine 1.09. The pattern black lines in the registration image for the Kinect

v2 are probably from the morphological transformation of  $1920 \times 1080$  RGB image to  $512 \times 424$  depth image. It can be seen that Carmine 1.09 has a very small offset in registration, which can be tolerated since that error is averaged across multiple RGB-D images which are fused into a single model. Astra S has a considerable offset in the registered images which results in incorrectly textured 3D models. The detected offsets could be as a result of imprecise factory calibration or problems with synchronization between color and depth image streams, although synchronization option was turned on via OpenNI 2 API for both Astra and Carmine camera.

In this article, primarily factory registration is used unless otherwise specified. In order to possibly improve the results, we also performed a checkerboard calibration between the color and infrared cameras for Astra and Carmine. The calibration of Carmine camera did not produce any significant improvement over the factory calibration. However, in the case of Astra sensor, calibration did produce significant improvement, as it can be seen in the fourth column of Fig. 2. It must also be mentioned that even after the calibration of Astra S, a constant offset between the color and depth image was noticed. Only after shifting the depth image 7 pixels to the left did the registration improve. This observation indicates that the registration of the color and depth images can be further improved by conducting calibration between the coordinate systems of the color and depth images instead of just color and infrared image, which were conducted in this paper. There are numerous articles on the topic of calibration of color and depth cameras, some of which are [19, 20]. Registration of color and depth images, as presented in the aforementioned research papers, would require considerable additional effort and is therefore out of the scope of this paper.

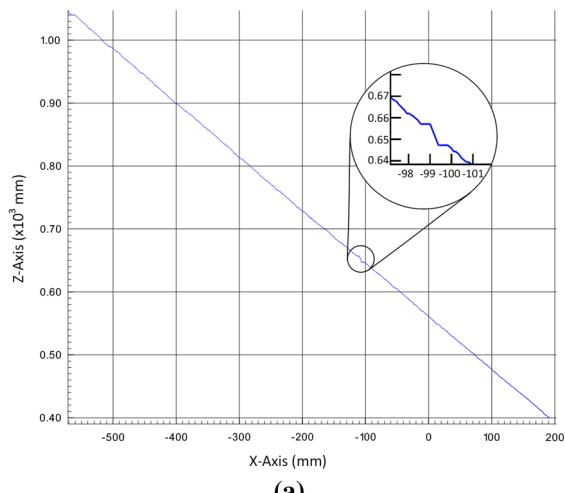
Besides the problem with registration, Astra has an anomaly with its depth measurement. Astra S camera has a measurement discontinuity which, in our case, appears typically between 645 and 655 mm. Figure 3a shows a single row from the depth image of a flat wall recorded at an angle. The surface distortion is better perceived on the point cloud displayed in Fig. 3b. Since there are multiple discussions in official Orbbec community about similar discontinuities, this anomaly does not affect only the particular instance of the camera used in the experiments reported in this paper.

It is not certain whether this is a hardware or firmware error, but we hope it will be resolved in future revisions of the Astra S camera.

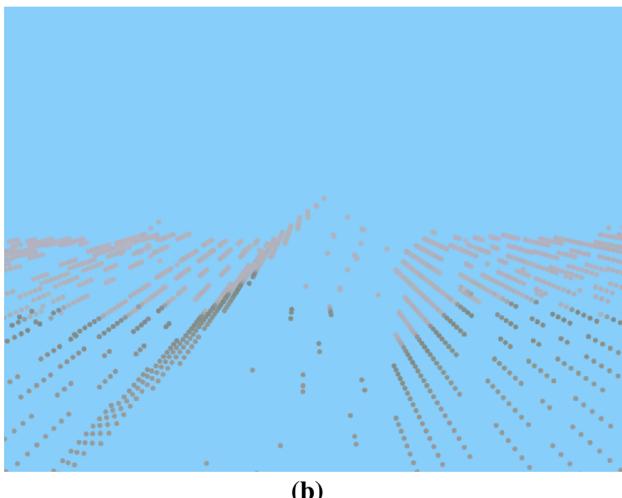
Since all considered cameras are based on light projection, depending on whether it is a specific pattern in the case of Astra and Carmine or light pulses in the case of Kinect v2, sensitivity to reflective surfaces in the recording environment must be considered. Figure 4 shows how partially glossy side of a DVD disk and a metallic case affect depth measurement.



**Fig. 2** Factory RGB to depth registration image and point clouds for Astra S (first column), Carmine 1.09 (second column), Kinect v2 (third column) and manual calibration for Astra S (fourth column)



(a)



(b)

**Fig. 3** Orbbee Astra S depth measurement discontinuity detected on a depth image of a flat wall recorded at an angle, **a** single row from the depth image, **b** surface distortion

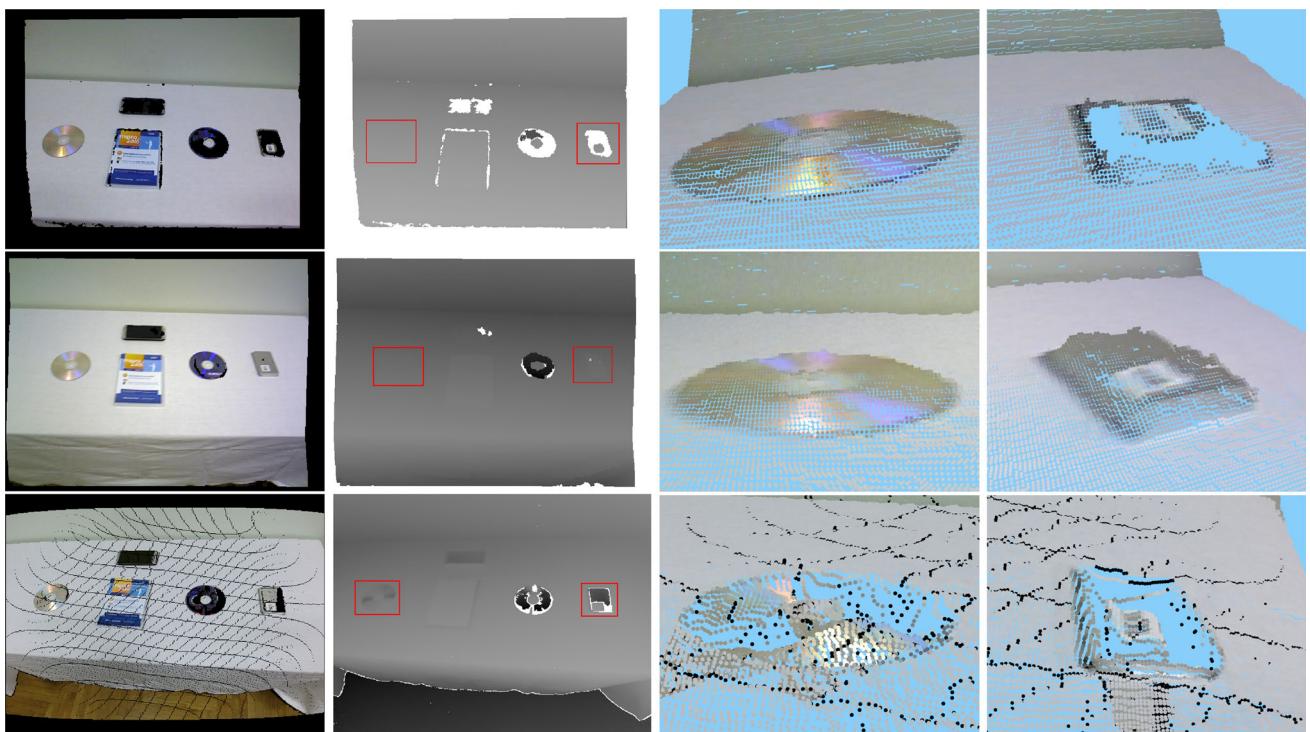
**Table 2** Number of sub-images generated depending on the size of  $w$

$w$	Initial reference database		Test images	
	$T_{sub}$	$W_{sub}$	$T_{sub}$	$W_{sub}$
16	40,200	430	56,280	715
32	9900	101	13,860	226
64	2400	26	3360	72

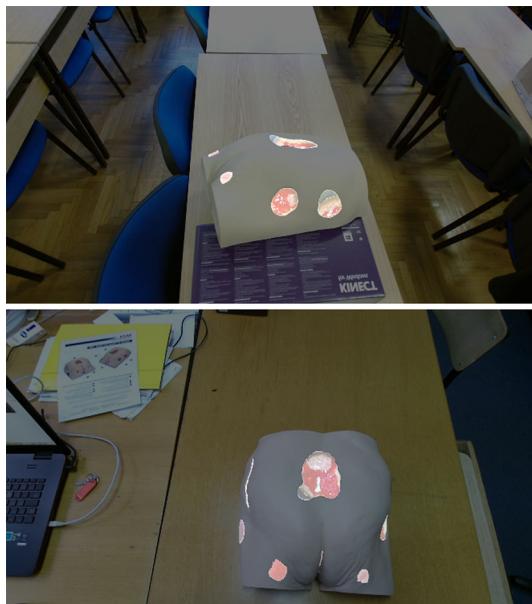
As can be seen, Astra is more susceptible to surface reflectivity than Carmine, which results in some surfaces not being seen in the depth images. The Carmine camera is less susceptible to reflective surfaces, which can almost be fully reconstructed, but those surfaces have more measurement noise. The Kinect v2 is very susceptible to reflectivity and although the reflective surfaces are reconstructed, i.e., they have depth measurement, those measurements are placed much further away than they really are which results in considerable distortion of surfaces. The reflective side of the DVD disk causes extreme measurement error for all considered cameras. Therefore, when using these types of cameras for analysis of wounds with glossy parts, measurement errors can be expected.

#### 4 Wound detection

In order to identify the relevant part of the scene whose accurate 3D model, we want to obtain, a wound must be detected in the input RGB-D image. Since 3D reconstruction is performed in real time with a moving handheld camera, wound detection needs to be fast and robust. The wound detection subsystem presented in this section primarily detects and marks regions on the image as potential wounds and



**Fig. 4** The first column are color to depth registered images, the second column are depth images and the third and fourth column are point clouds representing a DVD and a metallic case objects. The first row is obtained by Astra S, the second by Carmine 1.09 and the third by Kinect v2 camera



**Fig. 5** Ground truth image superimposed on the original image. Wound regions are represented by higher intensity pixels

allows the user to verify the position of the wound. Thus, even though the system is allowed to detect several possible regions and mark them as wounds, some of which might be falsely classified, the system *must always* detect the region where the wound is located.

The applied wound detection approach is based on color histograms and  $k$ -Nearest-Neighbor algorithm ( $k$ NN). Color histograms of a potential regions are created, and each region is classified using a  $k$ NN algorithm. For the purpose of classification, a database is created against which all future images are compared and classified. This database is herein referred to as the reference database.

A total of 12 images of the Seymour II wound care model by VATA Inc were generated under different lighting conditions and surroundings. All images were of dimension  $1920 \times 1080$  pixels. Ground truth images are generated by manually labeling all wound pixels in the considered 12 images. Five images were selected and used to create the reference database, while the remaining images were used as test images. Using a uniform grid of  $w \times w$  square cells or blocks, sub-images are generated for all images in the reference database as well as for the test images. Each sub-image is characterized by a feature vector obtained by concatenating the 1D histograms of the H and S color channels in the HSV color space. Each sub-image in the reference database is classified as being a wound if more than 50% of the pixels in the corresponding ground truth sub-image are labeled as wound pixels; otherwise, it is classified as not-wound. The created color histograms with assigned class represent entries in the reference database. The considered classifier is evaluated using the test images. Each sub-image or image block

of the test image is classified using  $k$ NN algorithm and the reference database.

Figure 5 shows an example of ground truth label image superimposed on the original image. Various values of  $w$  (16, 32 and 64) and various values of the number of bins ( $nbins = 32, 64, 96, 128$ ) of the 1D histograms were tested for the purpose of this application.

The number of sub-images in the reference database and those generated as test images depends on the value of  $w$ . An overview of the number of generated sub-images is provided in Table 2, where  $T_{sub}$  represents the total number of sub-images and  $W_{sub}$  the total number of sub-images classified as wound.

It can be noticed from Table 2 that out of the total number of sub-images generated, only about 1% consists of wound sub-images. Due to the highly skewed nature of the data in the initial reference database, the size of the majority class (non-wounds) was reduced in the training set, i.e., the original reference database was “under sampled”. 5  $W_{sub}$  instances of non-wound sub-images were randomly selected without replacement. Thus, the final reference database had a total of 6  $W_{sub}$  sub-images. The choice of the ratio of non-wounds to wounds in the reference database might be an interest of future research. At this stage, the 5 to 1 ratio was arbitrarily chosen so as to satisfy the following conditions: increase the total percentage of the wound class in the reference database; still have a relatively larger percentage of non-wound class since this class obviously has a much more diverse feature set.

Suitable values of  $k$  for the  $k$ NN algorithm for each combination of  $w$  and  $nbins$  were determined using 10-fold cross validation on the reference database. The results of the cross validation procedure are presented in Table 3 (where  $err_{avg}$  represents the average number of wrongly classified sub-images during the cross validation procedure).

Analyzing Table 3, the best results are obtained for  $nbins = 64$  irrespective of the value of  $w$ , with  $k = 10, 5$  and 11 for  $w = 16, 32$  and 64 respectively. The overall best result is obtained for  $w = 64$ . Each of the aforementioned combinations of  $w$ ,  $nbins$  and  $k$  represent a potential classifier. The quality of each of these 3 classifiers was ascertained by testing on the entire reference database and calculating

**Table 3** Cross validation results obtained for different combinations of  $w$  and  $nbins$

$nbins$	$w = 16$		$w = 32$		$w = 64$	
	$err_{avg}$	$k$	$err_{avg}$	$k$	$err_{avg}$	$k$
32	7.5	9	1.6	4	0.7	9
64	6.8	10	1.3	5	0.4	11
96	7.3	3	1.4	3	0.6	13
128	7.3	12	1.6	3	0.9	20

the recall and total percentage error values (Tables 4, 5). The recall value,  $R$ , is defined by:

$$R = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \quad (1)$$

where *true positives* refers to the number of correctly classified wound sub-images, whereas *false negatives* refers to the number of wound sub-images classified as non-wound. The total percentage error,  $err\%$ , is defined by:

$$err\% = \frac{\text{false positives} + \text{false negatives}}{\text{total number of sub images}} \times 100\% \quad (2)$$

where *false positives* refers to the number of non-wound sub-images classified as wound.

Analyzing the values obtained in Table 4, the first two classifiers ( $w = 16, nbins = 64, k = 10$ ) and ( $w = 32, nbins = 64, k = 5$ ) have comparable performance especially when considering their recall values. As stated at the beginning of this section, the system *must always* detect the region where the wound is located. This can be achieved only if the system has a recall value of 1. Thus, it can be assumed that a classifier with a fairly high recall designed on sub-images should be able to detect the position of the wound on an image.

All three classifiers in Table 4 above were tested on the entire test data set and the results provided in Table 5. The results show that the classifier ( $w = 16, nbins = 64, k = 10$ ) outperforms the other 2 classifiers in terms of the recall value and was selected as the most suitable.

Figure 6 shows an example of wound detection for Carmine 1.09 and Kinect v2 RGB-D camera, where all stages of the detection subsystem (classification, thresholding, representation) can be seen. The CUDA implementation of the detection subsystem have an average processing time of 115 ms for an  $640 \times 480$  image (Carmine 1.09, Astra S) and 95 ms for an  $512 \times 424$  image (Kinect v2) when using  $w = 16, nbins = 64, k = 10$  and the database with 2580 samples.

## 5 Wound reconstruction

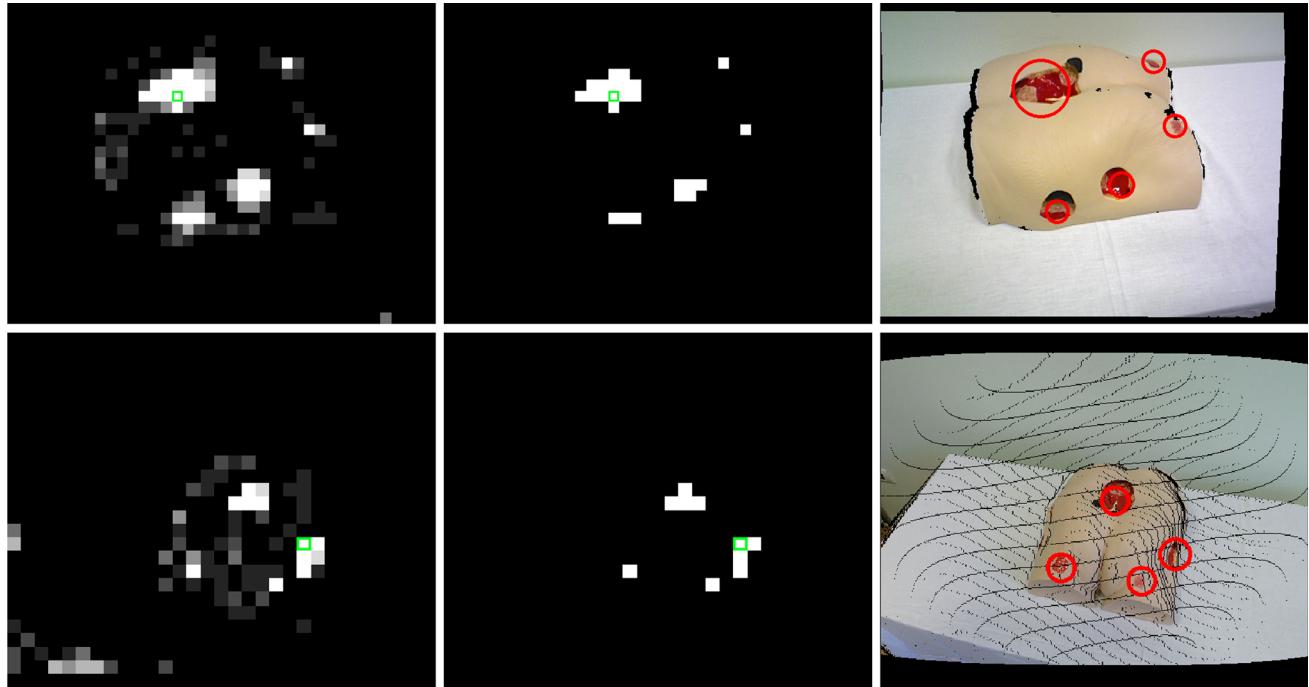
The methodology used in the proposed system for 3D reconstruction of wounds is basically the same as the one used in the KinectFusion system [1]. However, for the considered application, models in a higher resolution than that provided by KinectFusion are required. Since the source code of KinectFusion is not publically available, we could not adapt it for our purpose and therefore we developed our own 3D reconstruction subsystem. The 3D reconstruction subsystem is divided into two phases: scene fusion and mesh generation.

**Table 4** Comparison of the performance of the 3 best classifiers obtained in Table 3 on the reference database

Classifier	true negatives	false positives	true positives	false negatives	err%	R
$w = 16$	2122	28	394	36	2.48	0.92
$nbins = 64$						
$k = 10$						
$w = 32$	500	5	93	8	2.15	0.92
$nbins = 64$						
$k = 5$						
$w = 64$	125	5	23	3	5.13	0.88
$nbins = 64$						
$k = 11$						

**Table 5** Comparison of the performance of the 3 best classifiers obtained in Table 3 on the test dataset

Classifier	true negatives	false positives	true positives	false negatives	err%	R
$w = 16$	53612	1953	561	154	3.74	0.78
$nbins = 64$						
$k = 10$						
$w = 32$	13211	423	143	83	3.65	0.63
$nbins = 64$						
$k = 5$						
$w = 64$	3116	172	29	43	6.40	0.40
$nbins = 64$						
$k = 11$						



**Fig. 6** Applying classifier ( $w = 16$ ,  $nbins = 64$ ,  $k = 10$ ) on test image from Carmine 1.09 (first row) and Kinect v2 (second row). The first column shows the result of per block classification obtained using kNN and reference database showing the probability of a block being a wound, the second column shows the result of thresholding the images

the from first column, the third column shows potential wound regions drawn on the input image. For the purpose of clarity, a single image block in the first two columns is denoted by green square outline (color figure online)

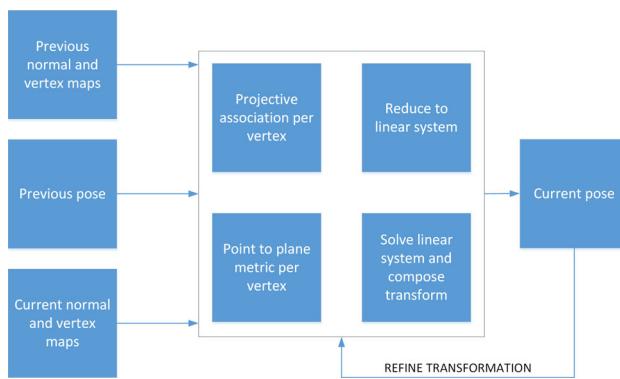
## 5.1 Scene fusion

Scene fusion is the process where each captured RGB-D frame is integrated into a reconstructed volume. This process is further divided into two sub-phases: registration and integration. Registration determines the rigid body transformation between frames and is based on the Iterative Closest Point (ICP) algorithm [2]. ICP is a widely used algorithm for determining rigid body transformation of objects comprised of point clouds or meshes. Input to ICP are typically two point clouds or mesh samples and the initial guess of their relative rigid body transformation. The ICP then iteratively refines the transformation by repeatedly generating corresponding pairs of closest points and minimizing an error metric.

A slightly modified ICPCUDA implementation [21] is used for frame to model registration. A modification to the implementation was introduced only to enable the use of Kinect v2 data properly. The ICPCUDA work diagram can be seen in Fig. 7. The registration sub-phase performance was on average 8.73 ms per frame for Kinect v2 depth images and 13.1 ms for Astra and Carmine depth images.

The integration sub-phase is based on a truncated signed distance function (TSDF). Signed distance function (SDF) is a nonparametric surface representation often used in computer graphics. It was introduced in [22], primarily for fusing depth maps. It is based on discretizing the volume into voxels of the same size. Each voxel is assigned an SDF value. Voxels with a positive SDF value are considered to be outside the object surface, while those with negative values are considered to be inside the object. Therefore, the iso-surface corresponding to value 0 represents the object surface. By truncating the SDF value to the interval  $[-1, 1]$ , the TSDF is obtained.

Because of the ray tracing nature of the integration process, undermining and tunneling cavities in wounds, which are included in the used wound care model, are currently not supported by the developed system because some of the voxels in the cavity will simply be filled by negative values since



**Fig. 7** ICPCUDA work diagram

they will be located under the surface when viewed from a different angle.

The integration sub-phase performance when using Kinect v2 was on average 8.25 ms per frame for a  $256 \times 256 \times 256$  volume of voxels and with color integration enabled, the performance was on average 14.03 ms. When using Astra and Carmine cameras performance per frame was 12.1 ms with color integration enabled and 7.722 ms when not.

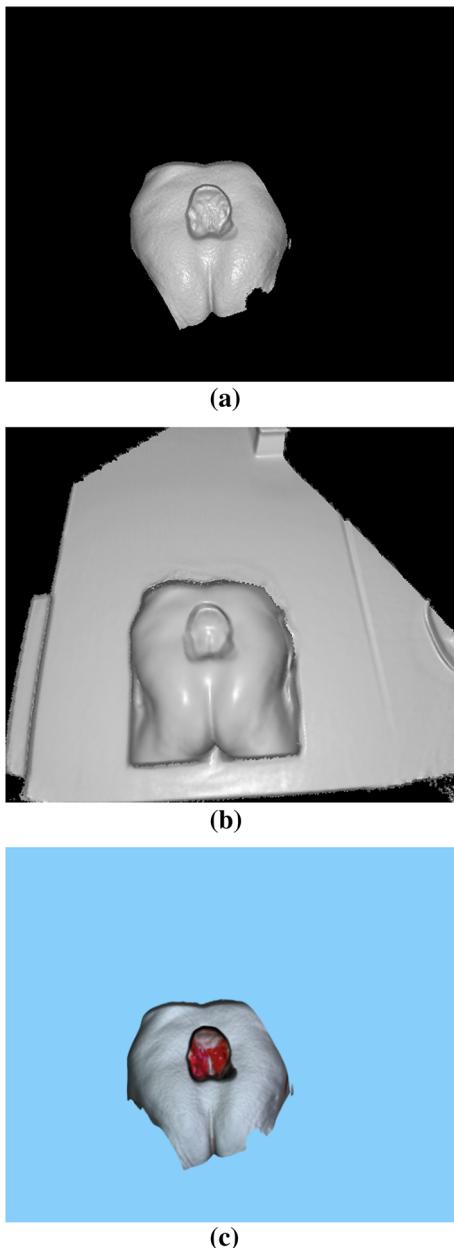
Visualization of the state of wound reconstruction during recording is provided by ray casting. Ray casting generates a rendered image from a given camera position by casting rays directly onto the volume, i.e., calculating the depth values of the zero crossing of the iso-surface of TSDF for rays originating from the camera position for each pixel of the rendered image. Depth images rendered by ray casting are also used in our system in a similar way as in KinectFusion, as a model for current pose estimation in the registration sub-phase.

Although a KinectFusion-based system is able to fuse partial views, a problem can arise in true medical applications, where large portions of the available geometry lack details that aid registration. In order to cope with this problem and keep the reconstruction precision, the developed system uses two TSDF volumes in parallel (where KinectFusion system uses only one), one high-resolution model with voxel size of 1 mm resulting in a total volume of  $0.25 \text{ m}^3$  (e.g., Fig. 8a), and a second lower resolution model with 4 mm voxel size and  $1 \text{ m}^3$  total volume (e.g., Fig. 8b). The low-resolution TSDF volume is used for model depth image rendering which is used for camera motion estimation. This TSDF volume contains a relatively large part of the scene, thereby taking into consideration cues from the surrounding area as well, which improves stability of camera motion estimation. It is represented in a lower resolution in order to save memory. The high-resolution TSDF volume is used for real-time visualization and later for 3D mesh generation of the wound model which is used for segmentation and measurement. The performance of TSDF ray cast rendering was on average 3.47 ms per frame for image resolution  $512 \times 424$  used by Kinect v2 and 5.28 ms for  $640 \times 480$  resolution used by Astra and Carmine.

## 5.2 Mesh generation

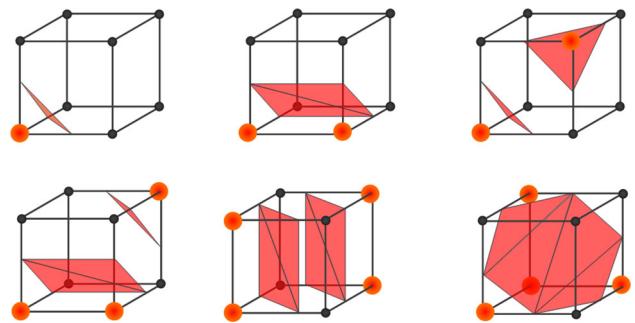
Given the TSDF volume gained by the scene fusion phase, there are two ways of obtaining a view or a rendering of the surface. The first option is to use an iso-surface generation algorithm such as the Marching cubes algorithm which returns a connected surface comprised of vertices and triangles (Fig. 8c). The other option is to use ray casting from some camera position directly onto the volume as explained earlier (Fig. 8a, b).

The Marching cubes algorithm is used for generating a high-resolution 3D model which is the result of the system



**Fig. 8** TSDF surface rendered from same viewpoint at iso-value 0 by **a** ray casting of the high-resolution TSDF, **b** ray casting of the low-resolution TSDF volume, **c** marching cubes algorithm applied to the high-resolution TSDF volume

presented in this paper. Marching cubes is an algorithm for polygonization of 3D scalar field at a certain level, i.e., iso-surface construction. It was originally developed by Lorensen and Cline for visualization of CT and MRI scans [23]. Today, it is a standard algorithm for surface reconstruction. The algorithm is based on analyzing each voxel (element of a 3D scalar field) independently and determining which voxel vertex is within the predefined surface border value. As a result, for a given voxel, the algorithm determines which triangles to create from a finite number of combinations. Since there are



**Fig. 9** Six out of 256 possible triangle combinations for marching cubes algorithm

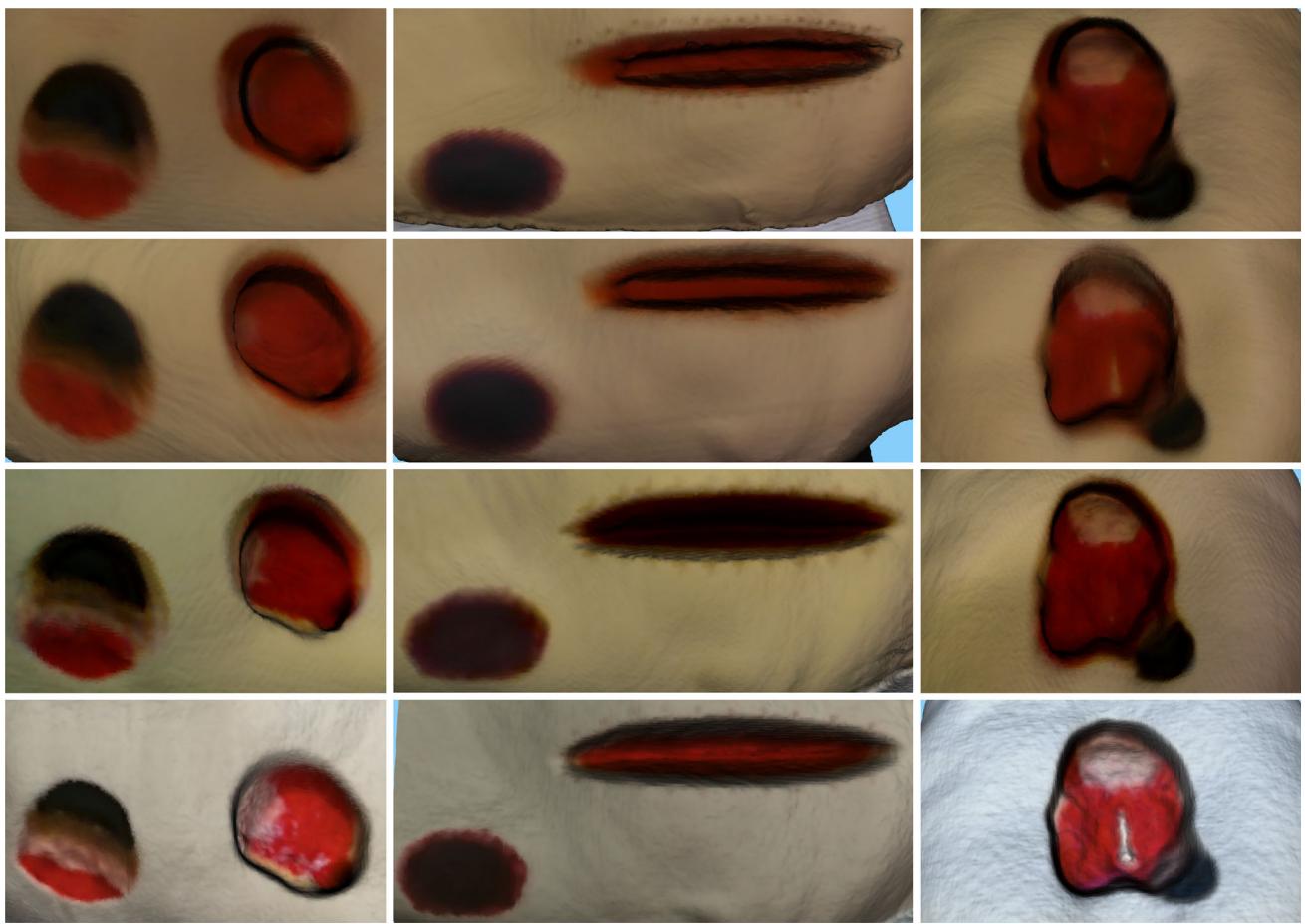
eight vertices per voxel, there are  $2^8 = 256$  possible triangle combinations. All 256 combinations are predefined in the implementation code which means that determining which triangle to create is very fast. Some of the possible triangle combinations within a voxel can be seen in Fig. 9, where the orange spheres are vertices having negative values of TSDF, while black spheres represent vertices having positive TSDF values.

Marching cubes CUDA implementation performance was on average 125 ms for a  $256 \times 256 \times 256$  TSDF volume which typically resulted in 3D models with about 325k triangles. Figure 10 displays reconstructed high-resolution colored 3D models of the wounds on the Seymour II wound care model by VATA Inc. for each of the three cameras.

As can be seen, the wounds reconstructed by Carmine show slight color smearing at the edges of the wound resulting from the less-than-perfect color to depth image registration. Although manually calibrated Astra S camera produces similar registration results as Carmine, because of the poor synchronization between depth and color streams, Astra S has a much larger color smearing problem around the wounds. Due to color smearing problems, a lot of details can be missed, such as the vertebra in the stage IV pressure ulcer wound shown in the third column in Fig. 10. Kinect v2 color to depth registration is superior than the other two cameras, which results in a sharper texture where a lot of details including the vertebra are discernable. However, because of the Kinect v2 sensitivity to reflective surfaces, a notable sinking of wound bottom surface geometry can be seen on all wounds since the wound bottom surface material is partially glossy. Due to that effect, measured volume of those models tends to be larger than the ones created by the Astra S and Carmine 1.09 cameras.

## 6 Wound segmentation

The approach applied in our system to segment out wounds is based on the assumption that a wound can be detected as a



**Fig. 10** High-resolution colored models with an average of 325k triangles when using Astra S (first row), manually calibrated Astra S (second row), Carmine 1.09 (third row) and Kinect v2 (fourth row) camera

region consisting of bumps and dents, characterized by high local surface curvature, surrounded by a smooth skin surface. Furthermore, if a wound is very shallow and cannot be distinguished from the surrounding skin using its geometric properties, the color of the wound can be used to segment it out from the uniformly colored surrounding skin. The proposed approach consists of three stages: (i) oversegmentation to surfels, (ii) grouping of surfels into smooth surface segments and (iii) heuristic selection of segments which represent wounds. The following subsections describe each of these three stages.

### 6.1 Surfels

Oversegmentation of images and point clouds into small groups of neighboring points, where each group represents a patch of an object surface, is a widely used preprocessing step in computer vision. These groups, referred to as superpixels [24,25], supervoxels [26] or surfels [27], provide more stable surface normal information and reduction of computational effort for subsequent processing stages. In our approach, the

mesh obtained by the 3D reconstruction subsystem described in Sect. 5.2 is segmented into approximately planar surface patches of approximately uniform color. These patches are referred to in this paper as surfels. Since the probability that a wound boundary lies in the middle of an approximately planar and uniformly colored surface patch is very low, it can be assumed that a complete surfel belongs to either a wound region or a skin region. Hence, the wound segmentation can be performed on the surfel level.

There are several oversegmentation methods which can be used as the preprocessing step in a wound segmentation process. The method used in the presented system for segmentation into surfels is similar to the region growing method proposed in [28], which segments a triangular mesh into approximately planar surfaces. The difference between this method and the method used in this paper is that we also use vertex color information in order to obtain approximately uniform colored surfels. This is important for detection of shallow wounds which can be segmented only according to their color. An advantage of our approach in comparison with the popular approach proposed in [26], where the total num-

ber of surfels is predefined and the surfel size constrained, is that our approach adapts the size and the number of surfels to the scene geometry. Planar surfaces or surfaces with large curvature radius in the scene are represented with a few large surfels, while highly curved details in the scene are represented with many small surfels, thus allowing precise estimation of wound size and shape.

The input to the segmentation procedure is a triangular mesh  $\mathcal{M}$ , which can be represented by a pair  $\mathcal{M} = (\mathcal{V}, \mathcal{T})$ , where  $\mathcal{V}$  is a set of vertices and  $\mathcal{T}$  is a set of triangles. Each vertex  $P_i \in \mathcal{V}$  is assigned a vector  $\mathbf{p}_i \in \mathbb{R}^3$  representing the position of the vertex relative to the mesh coordinate system, a unit normal vector  $\mathbf{n}_i \in \mathbb{R}^3$  perpendicular to the local surface and a color vector  $\mathbf{c}_i \in [0, 255]^3$ , whose elements represent the red, green and blue component of the vertex color normalized to the interval  $[0, 255]$ .

The applied segmentation process uses the neighborhood relations between mesh vertices. Two mesh vertices are neighbors if they belong to the same mesh triangle. The set of all neighbors of a vertex  $P_i \in \mathcal{V}$  is denoted in this section by  $\mathcal{N}(P_i)$ . Furthermore,  $\mathcal{N}(S)$ , where  $S$  is a set of mesh vertices, denotes the set of all vertices  $P_j \in \mathcal{V}$  such that  $\mathcal{N}(P_j) \cap S \neq \emptyset$ .

Surfels are detected by a region growing process starting with a randomly selected mesh vertex  $P_i$ , which is not already assigned to an existing surfel. A new surfel  $S_k$ , initially consisting of a single vertex  $P_i$ , referred to in this section as the *surfel center*, is recursively grown by adding neighboring vertices  $P_j \in \mathcal{N}(S_k)$  which satisfy a given criteria. A vertex  $P_j$  is added to the surfel  $S_k$  if: (i)  $P_j$  lies on the plane, defined by  $P_i$  and its normal  $\mathbf{n}_i$ , within a user-defined tolerance  $\tau_p$ , (ii) the difference between the normals  $\mathbf{n}_i$  and  $\mathbf{n}_j$  is smaller than a user-defined tolerance  $\tau_n$  and (iii) the difference between the colors of vertices  $P_i$  and  $P_j$  is smaller than a user-defined tolerance  $\tau_c$ . These three conditions are mathematically formulated by the following three equations, respectively

$$\max \{|\mathbf{n}_i^T(\mathbf{p}_j - \mathbf{p}_i)|, |\mathbf{n}_j^T(\mathbf{p}_i - \mathbf{p}_j)|\} \leq \tau_p \quad (3)$$

$$\|\mathbf{n}_j - \mathbf{n}_i\| \leq \tau_n \quad (4)$$

$$\|\mathbf{c}_j - \mathbf{c}_i\| \leq \tau_c \quad (5)$$

A vertex  $P_j$ , which satisfies these conditions, is added to the currently created surfel  $S_k$  if it was not previously assigned to another surfel. If  $P_j$  was previously assigned to a surfel  $S_l$ , then it is added to  $S_k$  only if the Euclidean distance between  $P_j$  and  $P_i$  is smaller than the Euclidean distance between  $P_j$  and the center of  $S_l$ . This procedure stops when all mesh vertices are assigned to surfels. The result of this procedure is a set of surfels  $\mathcal{S}$ . An example of segmentation into surfels is shown in Fig. 12b.

For each detected surfel  $S_k \in \mathcal{S}$ , the centroid  $\mathbf{m}_k$  and the mean color  $\mathbf{z}_k$  of the vertices  $P_i \in S_k$  is computed. Furthermore, the total least squares fitting of a plane to the vertices  $P_i \in S_k$  is performed. The unit normal vector of this plane is denoted in this section by  $\mathbf{u}_k$ . The computed vectors  $\mathbf{m}_k$ ,  $\mathbf{z}_k$  and  $\mathbf{u}_k$  are used in the surfel grouping process described in the following subsection. Algorithm 1 represents a pseudocode for the surfel segmentation procedure.

---

**Algorithm 1** Mesh segmentation into surfels

---

```

1:  $\mathcal{S} \leftarrow \emptyset$ 
2:  $\mathcal{V}' \leftarrow \mathcal{V}$ 
3:  $k \leftarrow 1$ 
4:  $\mathbf{a} \leftarrow$  vector of n elements  $a_i$  initially set to 0
5:  $\mathbf{d} \leftarrow$  vector of n elements  $d_i$  initially set to  $\infty$ 
6: repeat
7:    $P_i \leftarrow$  randomly selected vertex from  $\mathcal{V}'$ 
8:    $S_k \leftarrow \text{RegionGrowing}(\mathcal{M}, P_i, \text{SurfelCriterion},$ 
     $\text{UpdateSurfelMaps}(\mathbf{a}, \mathbf{d}))$ 
9:    $\mathbf{m}_k \leftarrow$  centroid of  $S_k$ 
10:   $\mathbf{z}_k \leftarrow$  mean color of vertices in  $S_k$ 
11:   $\mathbf{u}_k \leftarrow$  normal of the least squares plane fitted to  $S_k$ 
12:  Add  $S_k$  to  $\mathcal{S}$ 
13:   $\mathcal{V}' \leftarrow \mathcal{V}' \setminus S_k$ 
14:   $k \leftarrow k + 1$ 
15: until  $\mathcal{V}' = \emptyset$ 

```

---

The input to this algorithm is a mesh  $\mathcal{M}$  and thresholds  $\tau_p$ ,  $\tau_n$  and  $\tau_c$ . The result of the algorithm is a set  $\mathcal{S}$  of surfels  $S_k$ , representing disjoint subsets of  $\mathcal{V}$ . Each surfel  $S_k$  is assigned three vectors,  $\mathbf{m}_k$ ,  $\mathbf{z}_k$  and  $\mathbf{u}_k$ , which are previously defined. The algorithm uses two auxiliary vectors, *assignment map*  $\mathbf{a}$  and *distance map*  $\mathbf{d}$ . Element  $a_i$  of the assignment map  $\mathbf{a}$  represents the index of the surfel which vertex  $P_i$  is assigned to. Initially, all elements of  $\mathbf{a}$  are set to zero, which means that, at the beginning of the procedure, the mesh vertices are not assigned to any surfel. Procedure *RegionGrowing* represents a general region growing process defined on a graph. For the purpose of completeness, this procedure is presented as Algorithm 2.

The input arguments of *RegionGrowing* are a graph  $\mathcal{G}$ , the node  $P_{init}$  from which the region growing starts, the merging criterion *Criterion* and the procedure *UpdateMaps* which updates auxiliary maps needed to evaluate the region growing conditions. The region growing procedure examines the neighboring nodes of the nodes grouped in the growing region and joins a new node  $P'$  if it satisfies *Criterion*. If a new node is assigned to the grown region, procedure *UpdateMaps* is executed. The result of the discussed region growing algorithm is a subset of nodes  $R$ .

In Algorithm 1, *RegionGrowing* is applied to the mesh  $\mathcal{M}$ , representing a graph whose nodes are mesh vertices. *Criterion* used by *RegionGrowing* in Algorithm 1 is *SurfelCriterion*, which evaluates a candidate vertex  $P_j$  according to four conditions. The first three conditions are

**Algorithm 2** *RegionGrowing( $\mathcal{G}$ ,  $P_{init}$ , Criterion, UpdateMaps)*


---

```

1:  $Q \leftarrow$  empty queue
2: Put  $P_{init}$  into  $Q$ 
3:  $R \leftarrow \emptyset$ 
4: repeat
5:    $P \leftarrow$  the node at the top of  $Q$ 
6:   Remove  $P$  from  $Q$ 
7:   for every neighbor  $P'$  of  $P$  do
8:     if  $P'$  satisfies Criterion then
9:       Add  $P'$  to  $R$ 
10:      Insert  $P'$  at the bottom of  $Q$ 
11:      UpdateMaps
12:    end if
13:   end for
14: until  $Q$  is empty

```

---

defined by Eqs. (3), (4) and (5). If a vertex  $P_j$ , which is not assigned to any of the previously created surfels, satisfies these three conditions, it is joined to the grown surfel. In the case where a vertex  $P_j$  satisfies the three discussed conditions, but it is already assigned to a previously created surfel  $S_l$ , i.e.,  $a_j = l$ , the fourth condition is considered in order to decide whether  $P_j$  will be joined to the new surfel or it will remain in  $S_l$ . The vertex will be joined to the new surfel if its Euclidean distance to the initial vertex of the surfel is smaller than its distance to the initial vertex of  $S_l$ . The distances of vertices with respect to the initial vertex of the surfel which they are assigned to are stored in the distance map  $\mathbf{d}$ . Hence, the fifth condition can be formulated as

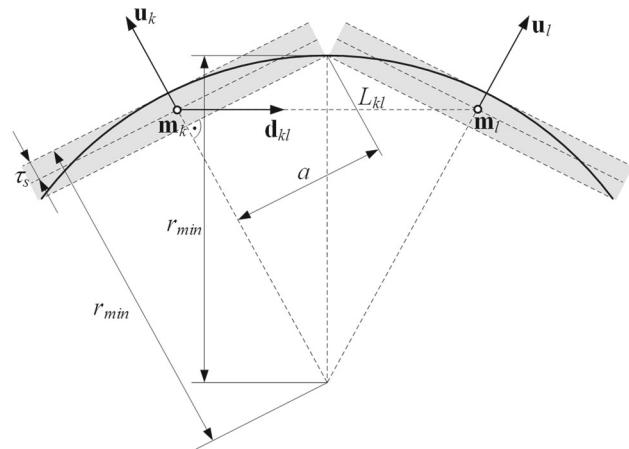
$$\|\mathbf{p}_j - \mathbf{p}_i\| < d_j. \quad (6)$$

If a new vertex  $P_j$  is assigned to the new surfel  $S_k$ , procedure *UpdateSurfelMaps* which updates maps  $\mathbf{a}$  and  $\mathbf{d}$  is executed. This procedure consists in setting  $a_j$  to  $k$  and  $d_j$  to  $\|\mathbf{p}_j - \mathbf{p}_i\|$ .

These parameter values were used in all experiments:  $\tau_p = 2.4$  mm,  $\tau_n = 0.44$ ,  $\tau_c = 20$ .

## 6.2 Segmentation to smooth surfaces

In [29], a method for segmentation of point clouds to smooth surfaces based on region growing is proposed. This method recursively groups points according to the similarity of normals of neighboring points and residuals in fitting a plane to the local surface. Residuals in plane fitting are used as a measure of the local surface curvature. Similarly to this method, in the wound segmentation subsystem presented in this paper, smooth surfaces are detected by region growing and local curvature is used as a cue for deciding on edges between segments. However, in our approach, segmentation to smooth surfaces is performed by surfel grouping. Furthermore, color information is also used in the grouping process.



**Fig. 11** Minimum sphere model

The detected surfels are grouped into smooth surfaces by a region growing process which uses neighborhood relations between surfels. Two surfels  $S_k$  and  $S_l$  are neighbors if there exist two points  $P_i \in S_k$  and  $P_j \in S_l$  which are neighbors, i.e.,  $P_j \in \mathcal{N}(P_i)$ . The region growing process starts by selecting a surfel, which currently does not belong to a smooth surface segment. This surfel is then used as the initial region, which is grown recursively by attaching neighboring surfels satisfying certain smoothness and color similarity conditions.

The geometric criterion used in region growing is based on the *minimum sphere model*. The minimum allowed curvature radius of the healthy skin surface is constrained to a predefined value  $r_{min}$ . Let us consider an ideal case where two neighboring surfels are detected on a spherical surface, as illustrated in Fig. 11, where the regions on a sphere corresponding to these two surfels are denoted by shaded rectangles. Furthermore, let us assume that the centroids  $m_k$  and  $m_l$  of these surfels lie approximately in the centers of these rectangles and that surfel normals  $\mathbf{u}_k$  and  $\mathbf{u}_l$  are identical to the local surface normal in these centroids. A property of surfels detected by the method described in Sect. 6.1 is that all points belonging to a surfel lie on a reference plane within a tolerance  $\tau_s$ . Let us consider the case, shown in Fig. 11, where the reference planes of two neighboring surfels are such that both surfels cover the maximum possible sphere regions. The distance between surfel centroid  $m_k$  and the surfel boundary, denoted by  $a$  in Fig. 11 is

$$a = \sqrt{r_{min}^2 - (r_{min} - 2\tau_s)^2}. \quad (7)$$

Hence, the projections of the normals  $\mathbf{u}_k$  and  $\mathbf{u}_l$  onto the straight line  $L_{kl}$  connecting the surfel centroids can be computed by

$$\alpha = \frac{a}{r_{\min}} = \sqrt{1 - \left(1 - \frac{2\tau_s}{r_{\min}}\right)^2}. \quad (8)$$

The value  $\alpha$  represents the estimated maximum value of the projection of the neighboring surfel normals onto the line connecting the surfel centers. This idealized model is used to formulate the criterion which two neighboring surfels must satisfy in order to be grouped in the same smooth surface. Let  $\mathbf{d}_{kl}$  be a unit vector parallel to the straight line  $L_{kl}$  connecting the centroids of two neighboring surfels  $S_k$  and  $S_l$ , i.e.,

$$\mathbf{d}_{kl} = \frac{\mathbf{m}_l - \mathbf{m}_k}{\|\mathbf{m}_l - \mathbf{m}_k\|}. \quad (9)$$

The projection of the surfel normal  $\mathbf{u}_k$  onto  $L_{kl}$  is

$$e_{kl} = \mathbf{u}_k^T \mathbf{d}_{kl}. \quad (10)$$

The projection  $e_{lk}$  of the surfel normal  $\mathbf{u}_l$  onto  $L_{kl}$  can be computed analogously. Our segmentation subsystem uses the following surfel grouping criterion: two surfels  $S_k$  and  $S_l$  are grouped into a smooth surface segment only if

$$\max\{|e_{kl}|, |e_{lk}|\} \leq \alpha + \sigma_n, \quad (11)$$

where  $\sigma_n$  is the tolerance introduced in order to compensate for the measurement noise.

Note that for any convex or concave surface,  $e_{kl} \cdot e_{lk} < 0$ . However, due to the measurement noise, for an approximately planar surface or a low curvature surface, the product  $e_{kl} \cdot e_{lk}$  can also have a small positive value. Therefore, in addition to condition (11), which is used if  $e_{kl} \cdot e_{lk} \leq 0$ , the following condition is used in the case where  $e_{kl} \cdot e_{lk} > 0$ :

$$\max\{|e_{kl}|, |e_{lk}|\} \leq \sigma_n. \quad (12)$$

Assuming that two adjacent surfels  $S_k$  and  $S_l$  lie on a spherical surface, all three vectors  $\mathbf{u}_k$ ,  $\mathbf{u}_l$  and  $\mathbf{d}_{kl}$  lie in the same plane, i.e., the plane defined by  $\mathbf{u}_k$  and  $\mathbf{d}_{kl}$  and the plane defined by  $\mathbf{u}_l$  and  $\mathbf{d}_{kl}$  are identical. However, in a real case, these two planes can differ due to the measurement noise. The final geometric criterion for grouping of two adjacent surfels is that the length of the vector representing the difference of the normals of these two planes is  $\leq \sigma_n$ , which can be formulated by

$$\left\| \frac{\mathbf{u}_k \times \mathbf{d}_{kl}}{\|\mathbf{u}_k \times \mathbf{d}_{kl}\|} - \frac{\mathbf{u}_l \times \mathbf{d}_{kl}}{\|\mathbf{u}_l \times \mathbf{d}_{kl}\|} \right\| \leq 2\sigma_n. \quad (13)$$

In addition to the described geometric criterion, average color similarity is used as a constraint for surfel grouping. Two neighboring surfels  $S_k$  and  $S_l$  are grouped into a smooth surface segment only if

$$\|\mathbf{z}_k - \mathbf{z}_l\| \leq \tau_z, \quad (14)$$

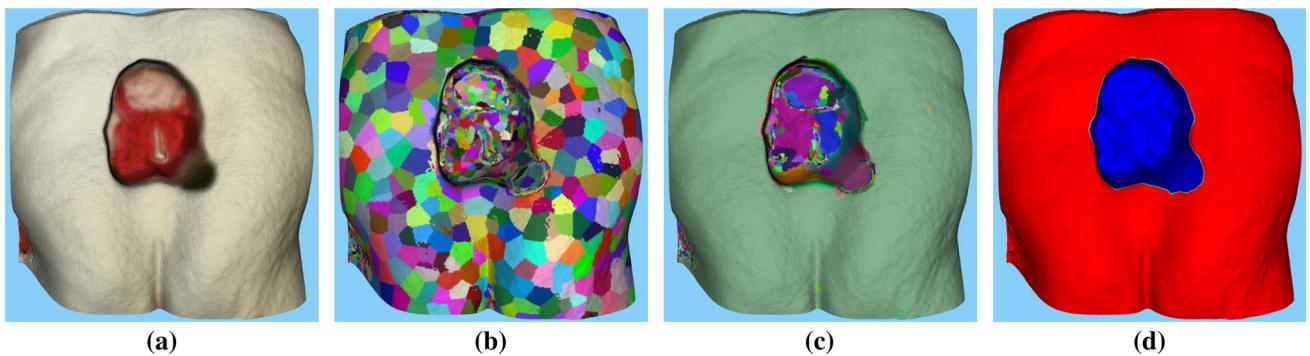
where  $\mathbf{z}_k$  and  $\mathbf{z}_l$  are the average colors of the surfels  $S_k$  and  $S_l$  and  $\tau_z$  is a color tolerance. The applied region growing procedure is analogous to the one used for segmentation into surfels. The only differences are that surfels are used instead of mesh vertices as elements which are aggregated into smooth surfaces and another criterion based on conditions (11), (12), (13) and (14) is used for region growing. An example of the described segmentation to smooth surface segments is shown in Fig. 12c, where each segment is represented by different color. The healthy skin is represented by a single green segment, while the wound region consists of multiple segments.

These parameter values were used in all experiments:  $r_{\min} = 50$  mm,  $\tau_s = 0.4$  mm,  $\sigma_n = 0.075$ ,  $\tau_z = 30$ . Colors are represented by values from the interval [0, 255] for each channel.

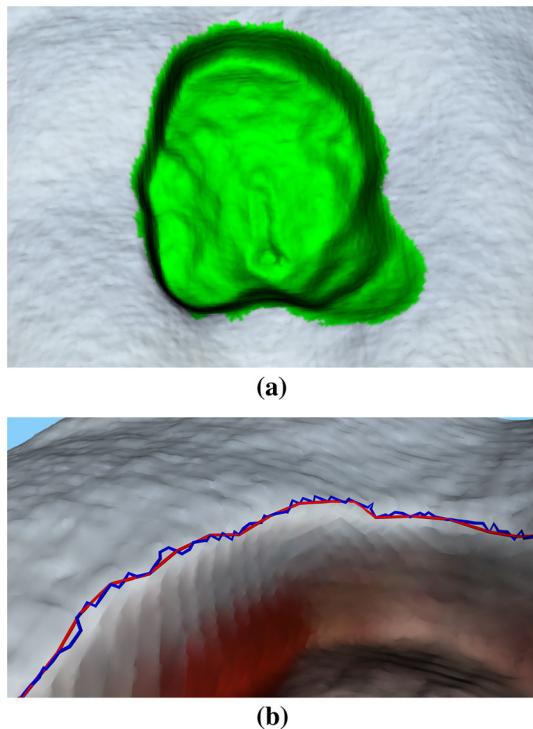
### 6.3 Segmenting wound from the surrounding healthy skin

The result of the surfel grouping procedure described in Sect. 6.2 is a set of smooth uniformly colored surface segments. Due to its irregular shape and non-uniform color, the wound region is expected to consist of multiple smaller regions surrounded by a large smooth surface segment representing the healthy skin. The heuristic wound segmentation approach proposed in this paper is based on two assumptions. The first assumption is that the skin region surrounding the wound region is the largest smooth surface segment in the mesh. The second assumption is that the segment closest to the wound center, determined by the wound detection approach proposed in Sect. 4, belongs to the wound region. Considering these two assumptions, the wound region is determined by a region growing procedure, which starts by creating the wound region, consisting initially of the single segment which is closest to the wound center, and grows this region recursively by adding neighboring segments until reaching the largest smooth mesh surface segment. The result obtained by applying this method is shown in Fig. 13a, where the wound region is represented by green color, while the surrounding skin is represented by light gray color. The performance of the whole segmentation procedure was on average 220 ms.

The mesh generated by the Marching cubes algorithm is comprised of very small and almost uniformly distributed triangles. The triangles are small due to the fact that the TSDF voxel size is 1 mm. Since the segmentation region growing process is driven by vertices as well as normal and color information, the resulting boundary between skin and wound could potentially be very jagged. In order to determine the volume of the wound, the surface of the healthy skin



**Fig. 12** **a** Reconstructed wound, **b** segmentation into surfels, **c** final regions obtained by region growing, **d** wound region after edge spline interpolation



**Fig. 13** **a** Extrapolated wound region; **b** boundary generated by wound region extrapolation (blue) and spline interpolation (red) (color figure online)

before injury needs to be reconstructed. This reconstructed skin surface covering the wound is referred to in this section as *virtual skin*. Virtual skin estimation is done using boundary information. The more jagged the boundary, the more likely the virtual skin surface geometry would be highly irregular. Therefore, the boundary is interpolated using spline in order to alleviate its jagged nature. As a result, the number of points defining the boundary is significantly reduced. Figure 13b shows an example of the original and the spline interpolated boundary. The spline interpolated boundary has only a sixth of the number of points of the original boundary.

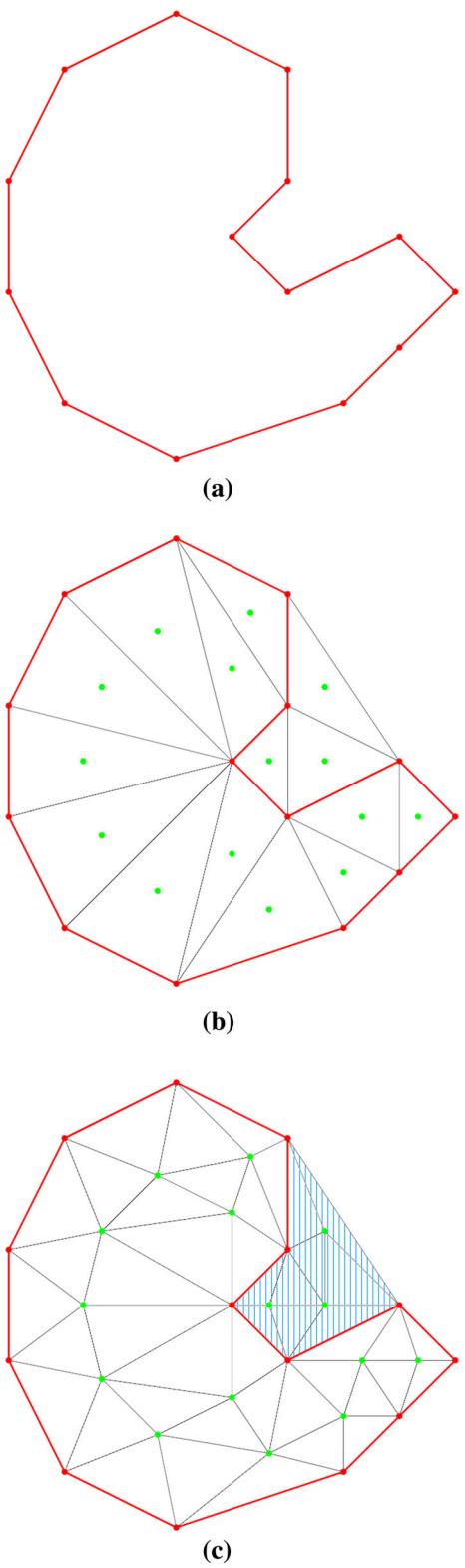
In order to isolate only the wound as a separate 3D model from the whole 3D scene generated, the Visualization Toolkit (VTK) [30] is used. The selection filter of the VTK library uses the spline interpolated boundary to generate scalar data (Fig. 12d) needed by the extraction filter, from the same library, to isolate the wound mesh into a separate 3D object.

## 7 Wound measurement

The developed system, uses the 3D wound model created by the approach described in Sect. 5 and the results of the segmentation presented in Sect. 6 to compute the perimeter, area and wound volume as end results. The perimeter is easily calculated as the length of the boundary of the wound region obtained by segmentation. In order to calculate the wound area and volume, a cover or a virtual skin needs to be generated.

When filming the wound using the RGB-D camera, the wound recording usually begins at an angle approximately perpendicular to the wound surface, and thus, the wound border is well defined on the 2D  $x$ - $y$  plane. This fact is exploited when generating the virtual skin. An iterative 2D Delaunay triangulation algorithm is used in virtual skin generation as seen on Fig. 14. The initial iteration of the Delaunay algorithm starts with an initial point set containing the points from the spline interpolated wound boundary (Fig. 14a). Each consecutive iteration expands this point set by including the centers of the Delaunay triangles of the previous iteration (Fig. 14b). Experiments indicate that three iterations are sufficient to generate a reasonable approximation of the virtual skin. Since the Delaunay algorithm creates a convex hull and the wounds do not always have convex shape, the triangles generated by this process which are outside the wound boundary are removed (Fig. 14c).

Figure 15 shows the reconstructed and isolated wounds and corresponding virtual skins of the Seymour II wound care model recorded using a Kinect v2 camera.



**Fig. 14** Example of 2D Delaunay virtual skin generation. **a** The initial point set which includes only points on the boundary, **b** Delaunay triangulation of the initial point set with green points representing triangle centers, **c** second iteration Delaunay triangulation with striped pattern triangles representing the triangles which will be removed from the final mesh of the virtual skin (color figure online)

The area and volume are calculated by appropriate algorithms implemented in the VTK library. The area is calculated as the sum of areas of individual triangles in the virtual skin mesh, while the individual triangle area is calculated using Herons formula:

$$s = \frac{a + b + c}{2}, \quad (15)$$

$$A_i = \sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - c)},$$

where  $a, b, c$  are lengths of the triangle sides.

The volume is calculated using a morphometric method called MUNC (Maximum Unit Normal Component) and a discrete form of the divergence theorem (DTA). The accuracy and precision of these methods usually used for area and volume calculation is described in [31]. The DTA algorithm requires the surface to be piecewise smooth and closed. The volume calculation method uses surface-based geometry in the form of the point list. MUNC basically defines which normal component is maximum at each point, and in this implementation, it is used to define the coefficients which distribute volume calculation across three cardinal directions. The discrete form of the divergence theorem is used to estimate the object volume from the list of the points on its surface. In our case, the points are represented by the centroids of the mesh triangles. Therefore, the volume is determined using the following equation:

$$V = k_x \cdot \sum_{i=0}^N (x_i \cdot n_{x_i} \cdot A_i) + k_y \cdot \sum_{i=0}^N (y_i \cdot n_{y_i} \cdot A_i) + k_z \cdot \sum_{i=0}^N (z_i \cdot n_{z_i} \cdot A_i), \quad (16)$$

where  $x_i, y_i$  and  $z_i$  are the coordinates of the  $i$ th triangle centroid,  $n_{x_i}, n_{y_i}, n_{z_i}$  are the components of the corresponding unit normal,  $A_i$  is the corresponding triangle area, while  $k_x$ ,  $k_y$ , and  $k_z$  are coefficients whose sum is equal to 1.

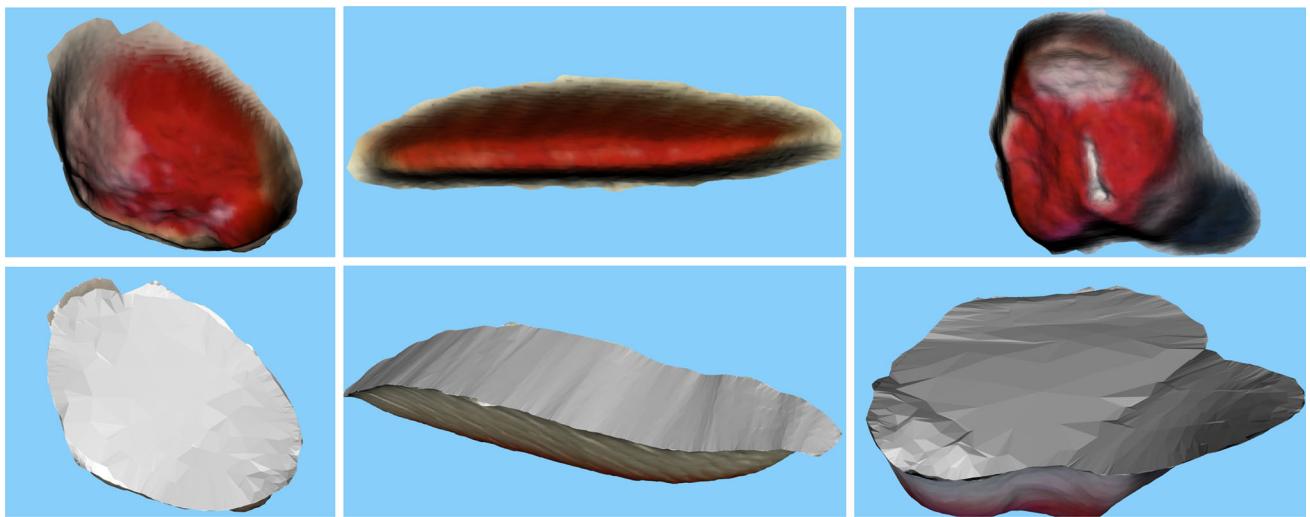
The coefficients  $k_x$ ,  $k_y$ , and  $k_z$  are calculated as a fraction of the total number of points in which the MUNC of these points is in the direction indicated by the subscript of the coefficient. Therefore:

$$k_x = \frac{1}{N} \cdot \left( N_{MUNC_x} + \frac{N_{xyz}}{3} + \frac{N_{xy} + N_{yz}}{3} \right),$$

$$k_y = \frac{1}{N} \cdot \left( N_{MUNC_y} + \frac{N_{xyz}}{3} + \frac{N_{xy} + N_{yz}}{3} \right), \quad (17)$$

$$k_z = \frac{1}{N} \cdot \left( N_{MUNC_z} + \frac{N_{xyz}}{3} + \frac{N_{xy} + N_{yz}}{3} \right),$$

where  $N$  is the total number of points (triangles),  $N_{MUNC_x}$ ,  $N_{MUNC_y}$  and  $N_{MUNC_z}$  are the numbers of points with corresponding maximum normal components in  $x$ ,  $y$ , and  $z$



**Fig. 15** Isolated 3D wound models with virtual skin generated

direction, respectively,  $N_{xyz}$  is the number of points whose all normal components are equal,  $N_{xy}$  and  $N_{yz}$  are the number of points whose normal components indicated by the subscripts are equal.

Each discrete sum in (16) estimates the entire volume of the object. Using the coefficients  $k_x$ ,  $k_y$ , and  $k_z$  provides a weighted sum of these three estimates thus providing a more accurate volume measurement. Calculating the coefficients in such a way minimizes the error incurred when many of the differential triangle surfaces have small cross sections. The area, volume and perimeter calculation requires on average 17 ms per 3D model.

## 8 Experimental results

### 8.1 Experimental setup

The system proposed herein was fully developed in C++ and CUDA v7.0. Furthermore, all experiments were run on a mid-range laptop computer with Intel Core i7 4710HQ CPU, Nvidia 860M GPU and Windows 8.1 Pro  $\times$  64 operating system. In addition to the enumerated RGB-D cameras and the PC computer, the used materials also included a Seymour II wound care model by VATA Inc. which was used for the experimental evaluation of the developed system. In total, there were 25 recordings used for Kinect v2 experiments and 30 for Astra and Carmine experiments. Furthermore, 27 recordings for Carmine and 25 for Astra were made when using manual calibration. The recording length was varied and therefore recordings from Kinect v2 contained between 150 and 346 frames, while Astra and Carmine recordings contained between 240 and 520 frames. The experiments

were performed under different illumination conditions: natural lighting, fluorescent lighting and LED lighting.

### 8.2 Measurement repeatability

The repeatability experiments demonstrate stability of measurement capability. Table 6 shows the average and coefficient of variability of the perimeter, area and volume measurements for the three largest wounds on the Seymour II wound care model. Low variability indicates robustness of the system.

In general, it can be seen that Kinect v2 has smaller coefficient of variability for perimeter and area measurements, while Astra S and Carmine 1.09 have smaller coefficient of variability for volume measurement. The color smearing on the surrounding healthy tissue of the 3D model, due to a poorer color to depth registration in Astra and Carmine cameras, can change the perimeter and consequently change the area measurement from sequence to sequence. Therefore, coefficient of variability is significantly higher for these two cameras and differs between wounds.

Conducting a manual calibration between color and infrared cameras can lead to a better registration and more consistent perimeter and area measurement. Astra S shows considerable improvement when using manual calibration, where it shows similar repeatability performance as Carmine 1.09 camera. Carmine 1.09 camera shows similar performance when using factory or manual calibration.

The volume measurement for all considered cameras can depend on the camera motion during the recording of a particular image sequence. In the case where a wound has cavities which extend below the skin, e.g., the wound on the Seymour II wound care model shown in Fig. 12, these cavities could be easily omitted in the recording process. The result

**Table 6** Measurement repeatability experiments

Wound type	Perimeter Avg (mm)	Variability (%)	Area Avg (mm <sup>2</sup> )	Variability (%)	Volume Avg (mm <sup>3</sup> )	Variability (%)
Orbbec Astra S						
Stage IV pressure ulcer	353.3/347.7	4.985/2.304	7128.2/6969.6	4.113/3.609	92748.1/89106.6	3.755/3.465
Stage III pressure ulcer	190.9/179.6	14.738/4.892	2205.3/2196.5	13.112/8.705	17172.8/17936.2	9.545/2.917
5 1/2" long dehisced surgical wound	263.4/256.7	3.557/3.292	2322.5/2351.2	7.154/5.062	20975.5/22742.8	6.742/5.137
PrimeSense Carmine 1.09						
Stage IV pressure ulcer	341.9/338.7	2.973/1.73	6986.1/6934.5	4.784/4.505	86550.4/87678.8	2.902/3.060
Stage III pressure ulcer	179.3/179.6	4.646/3.256	2291.3/2311.6	6.849/6.831	17357.8/17756.9	3.854/3.731
5 1/2" long dehisced surgical wound	269.3/267.6	2.322/3.458	2473.3/2459.6	9.684/7.501	20127.6/20728.3	1.376/1.977
Microsoft Kinect v2						
Stage IV pressure ulcer	354.2	2.608	7763.6	2.033	139074.3	4.885
Stage III pressure ulcer	192.1	2.707	2712.1	2.175	28511.9	12.875
5 1/2" long dehisced surgical wound	288.1	1.233	2898.0	1.294	39241.2	8.268

The results for Astra S and Carmine are shown in format: factory calibration/manual calibration

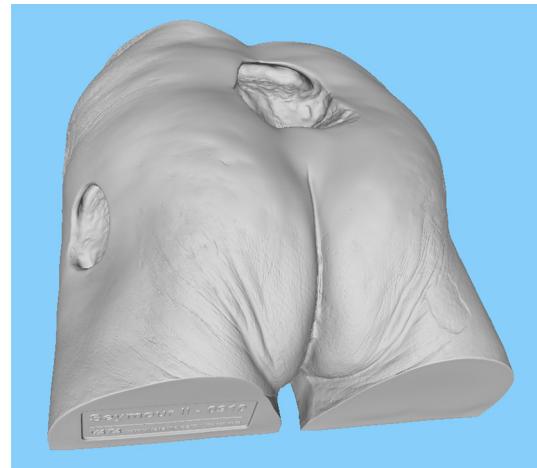
is narrowing of the base of the 3D wound model and consequently a smaller volume at certain sequences causing a larger coefficient of variability.

Regardless of the camera motion when recording an image sequence, due to the sensitivity to glossy surfaces, Kinect v2 camera adds error in depth measurement which could lead to sinking of the wound bottom or even distortion of the wound surface. This results in an increased coefficient of variability of volume measurement when using Kinect v2 camera.

### 8.3 Ground truth

The ground truth for assessing the accuracy of the proposed system was created by digitalization of the whole Seymour II wound care model using an industrial 3D scanning equipment based on GOM ATOS Triple Scan which has a precision in 10  $\mu\text{m}$  range. Glossy surfaces were treated by spraying dulling powder prior to scanning in order to remove reflectivity. The digitized model can be seen in Fig. 16.

The reconstructed model is near perfect and includes the subsurface cavities and tunneling parts of the wound. Since the developed system currently does not support such geometry as was explained in Sect. 5.1, those geometrical deformities were digitally filled prior to measurement. Besides our ground truth, another ground truth for Seymour II wound care model is reported in the literature, which is created by 3D CT-reconstruction with 0.625 mm slice using GE Healthcare device [15]. In [15], authors physically plugged cavities and holes prior to measurement and therefore our ground truth measurement should be similar; unfortunately, there are discrepancies with the volume measurement. Table 7 shows ground truth measurement by our approach and by [15] for the three wounds



**Fig. 16** Seymour II wound care model digitized by an industrial 3D scanner

analyzed in this research. Unfortunately, the 5 1/2" long dehisced surgical wound measurement is not available in [15]. As it can be seen, the ground truths are very similar for the parameter and area measurement, but the volume measurement shows difference of 10.7% for the larger wound and 9.4% for the smaller wound, even though cavities and holes were similarly plugged in both cases. The reasoning for those dissimilarities could possibly be in determining the border of the wounds where slight inconsistencies could result in large volume differences given the area of the wounds. For the sake of completeness, precision comparison in the next subsection will consider both ground truth measurements for the wounds where are available.

**Table 7** Ground truth measurement by our approach and by [15]

Wound type	Our GT measurement			GT measurement according to [15]		
	Perimeter (mm)	Area (mm <sup>2</sup> )	Volume (mm <sup>3</sup> )	Perimeter (mm)	Area (mm <sup>2</sup> )	Volume (mm <sup>3</sup> )
Stage IV pressure ulcer	334.9	7117.6	100821.2	343	6940	90000
Stage III pressure ulcer	175.7	2369.8	21542.8	170	2120	19500
5 1/2" long dehisced surgical wound	277.7	2400.0	28090.8	n/a	n/a	n/a

## 8.4 Accuracy

The reconstruction accuracy of the proposed system is evaluated by comparing the obtained measurement results to the ground truth generated for the purpose of this paper as well as to the ground truth used in [15]. Table 8 shows absolute percentile error between the average measurement of perimeter, area and volume presented in Table 6 and the ground truth measurements presented in Table 7.

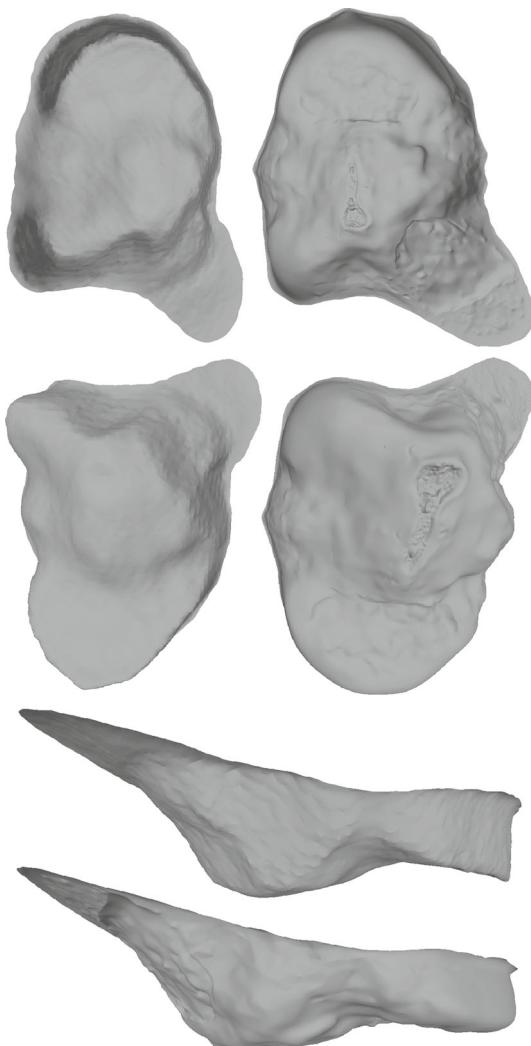
Astra and Carmine cameras show similar results in perimeter and area measurement error regardless of which ground truth is used with individual measurement being more accurate for one or the other camera. Overall, in perimeter and area measurement, Carmine is shown to be slightly more accurate than the Astra camera. The accuracy of the volume measurement depends on the ground truth which the measurement is compared to. Therefore, because of the smaller values of the volume in the ground truth used in [15], Astra and Carmine show smaller errors, which ranged from 0.992% to maximum of 11.93%. When the ground truth generated for this research is used, the error increases and ranges from 8.07% to a maximum of 28.34%. Overall, Astra is shown to provide slightly more accurate volume measurements than Carmine camera, regardless of the ground truth used. Furthermore, manually calibrating the Carmine camera produced similar results, while manually calibrating Astra camera resulted in overall smaller errors.

Microsoft Kinect v2 camera shows similar errors when measuring wound perimeter, but larger errors when measuring the area of the wound in comparison to Astra and Carmine cameras, regardless of the ground truth used. Unfortunately, Kinect v2 shows even larger error in volume measurement, which can be related to considerable sinking of the wound glossy surface, which, in the end, results in much larger volume measurements compared to the ground truth.

The research presented in [15], which uses a single RGB-D image for wound measurement, shows comparable measurement performance: the perimeter error was from 3.4 to 4.3%, the area error from 2.0 to 4.5%, while the volume measurement error was from 6.4 to 12.7%, when compared to the ground truth used in this research.

Volume measurement using noisy commodity cameras can easily be prone to large errors especially when recon-

structing larger depressions (wounds) in the surface. For example, if we consider stage IV pressure ulcer of the used medical model, with the surface area of approximately 7000 mm<sup>2</sup> and if the surface depth measurement of the surrounding skin is off by just 1 mm, it could potentially result



**Fig. 17** Comparison between the model of stage IV pressure ulcer generated by the developed system using Carmine 1.09 camera and the model from the ground truth generated by an industrial 3D scanner. The models are compared from the top, bottom and sideways. The model from the ground truth is on the right in the case of the top two pairs and on the bottom in the case of the last pair

**Table 8** Absolute percentile error of measurements compared to the ground truth in Table 7

Wound type	Our GT measurement			GT measurement according to [15]		
	Error perimeter (%)	Error area (%)	Error volume (%)	Error perimeter (%)	Error area (%)	Error volume (%)
<b>Orbbec Astra S</b>						
Stage IV pressure ulcer	5.505/3.834	0.149/2.087	8.007/11.61	3.014/1.382	2.712/0.418	3.053/0.992
Stage III pressure ulcer	8.635/2.246	8.027/8.394	20.28/16.74	12.27/5.674	4.024/3.608	11.93/8.019
5 1/2" long dehisced surgical wound	5.137/7.557	3.228/2.034	25.32/19.03	n/a	n/a	n/a
<b>PrimeSense Carmine 1.09</b>						
Stage IV pressure ulcer	2.104/1.128	1.848/2.572	14.15/13.16	0.306/1.259	0.663/0.078	3.832/2.718
Stage III pressure ulcer	2.077/2.262	4.442/3.594	19.42/17.57	5.499/5.691	8.078/9.038	10.98/8.938
5 1/2" long dehisced surgical wound	3.026/3.642	3.054/2.483	28.34/26.20	n/a	n/a	n/a
<b>Microsoft Kinect v2</b>						
Stage IV pressure ulcer	5.763	5.763	37.94	3.265	11.86	54.52
Stage III pressure ulcer	9.355	13.11	32.35	13.02	27.93	46.21
5 1/2" long dehisced surgical wound	3.749	20.75	39.69	n/a	n/a	n/a

The measurement results obtained by Astra S and Carmine are shown in format: factory calibration/manual calibration

in 7000 mm<sup>3</sup> of erroneous volume. The end result is the percentile error of 6.9 or 7.7% depending which ground truth is used. Considering the measurement noise of the considered RGB-D cameras, averaging nature of the TSDF and error of the ICP registration, the reconstructed surface could possibly have an error of more than 1 mm. A visual comparison between the reconstructed and automatically segmented stage IV pressure ulcer obtained by the developed system and the one from the ground truth is shown in Fig. 17.

## 9 Discussion and conclusion

In this article, a possibility of using RGB-D cameras for detection, modeling and measurement of chronic wounds is explored. Application of three different RGB-D cameras: Orbbec Astra S, PrimeSense Carmine 1.09 and Microsoft Kinect v2 is considered. All three cameras have their advantages and disadvantages. Kinect v2 is very sensitive to glossy and reflective surfaces but has the best factory color to depth registration. Astra and Carmine are much less sensitive to glossy surfaces, but their color to depth registration is less precise and typically requires manual calibration, furthermore Astra has certain depth measuring anomaly at a distance of about 645 mm which can create problems when fusing depth images.

A system for automatic measurement of wound parameters is presented, which could assist in monitoring wound healing process by enabling detailed wound analysis and, thereby, improve treatment for patients. The developed system consists of four subsystems: wound detection, 3D reconstruction, segmentation and measurement. The main goal of the detection subsystem is to find the center of the wound in order to facilitate the generation of high-resolution

local 3D models. The reconstruction system is based on KinectFusion, but it has some differences. For example, two separate TSDF volumes are used, volumetrically larger one for model depth image rendering and a smaller one for high-resolution 3D model generation. Segmentation is driven by both color and geometry in order to segment out wounds of various depths and sizes from the surrounding healthy tissue of a different color. The perimeter of the wound is measured by calculating the outline of the segmented wound, while area measurement is enabled by generating a virtual skin and volume measurement by creating a closed watertight surface.

The measurement repeatability experiments show that Kinect v2 provides more stable perimeter and area measurement because of better color to depth registration, whereas Astra S and Carmine 1.09 have more stable volume measurement because of less sensitivity to glossy surfaces. Although manual calibration of Astra and Carmine sensors does improve measurement stability, color to depth registration of Kinect v2 is still better.

Measurement precision is determined by comparing to two different ground truths, one created for the purpose of this paper and the other taken from the literature. The comparison shows high accuracy of the perimeter measurement for all three cameras, especially when compared to our ground truth, with Carmine 1.09 being better than the rest. Area measurement precision is clearly better for the Astra and Carmine cameras, which show similar results regardless of the ground truth used, whereas volume measurement seems to be problematic for all cameras, especially Kinect v2. Overall, Astra and Carmine cameras exhibit greater precision than Kinect v2 when comparing to the ground truth from the literature. Volume measurement is in general problematic in scanning depressions such as wounds, where small but sys-

tematic errors in depth measurement over large area could result in high volume measurement errors.

Overall, PrimeSense Carmine 1.09 has demonstrated to be the best solution for 3D reconstruction and measurement in a KinectFusion like system because of better color to depth registration than Astra S and better response to glossy surfaces than Kinect v2. Unfortunately, as mentioned earlier, since the acquisition of PrimeSense by Apple in 2013, their Carmine series of products as well as other products licensed by their technology have currently been discontinued. The alternative today are the Orbbec cameras that currently have some problems like subpar factory calibration and anomalies, which we hope will be resolved in future revisions.

In the future, we plan to continue developing the proposed system by developing a more comprehensive sample database for the detection subsystem, which should make it more precise. The reconstruction subsystem should enable support for wound cavities and the CUDA code should be more optimized, resulting in faster scene fusion. The segmentation could be improved by utilizing a different color space, like HSV or Lab instead of RGB, which should result in a more accurate segmentation. Accuracy of the proposed system could be improved by improving color to depth registration and using RGB-D cameras with more precise depth measurement. Furthermore, the system could be extended to provide information about the necrotic, fibrin and granulation tissue types on the wound surface. Finally, in order to get a better insight into practical applicability of the developed system, a more thorough testing on real wounds and different lighting conditions should be performed.

## References



**Damir Filko** received his B.Sc. degree in 2006 and Ph.D. degree in 2013, both in electrical engineering from Faculty of Electrical Engineering, Computer Science and Information Technology (FERIT Osijek), Josip Juraj Strossmayer University of Osijek, Croatia. He is currently employed as an Assistant Professor at Department of Automation and Robotics at FERIT Osijek. He (co-)authored more than 25 scientific papers published in proceedings of international conferences and journals.

His current research interests include application of computer vision in robotics and medicine.



Josip Juraj Strossmayer University of Osijek, Croatia. His main research interest is in robot vision.



**Emmanuel Karlo Nyarko** is currently an Assistant Professor at the Faculty of Electrical Engineering, Computer Science and Information Technology, Josip Juraj Strossmayer University of Osijek. He received his B.Sc., M.Sc. and Ph.D. degrees in 2001, 2005 and 2015, respectively, from the same institution, where he worked as a teaching and research assistant from 2001 to 2005. From 2005 to 2007, he was employed as a software developer in, Mono LLC, where his responsibilities included requirements analysis, design and development of various web and smart client applications. His research interests include robot vision, soft computing methods and machine learning.

## Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”). Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

[onlineservice@springernature.com](mailto:onlineservice@springernature.com)